TEMA 4: clasificación supervisada

José R. Berrendero

Departamento de Matemáticas, Universidad Autónoma de Madrid

Temas a tratar

- Planteamiento del problema
- La regla lineal de Fisher
- La regla de Mahalanobis
- Regla Bayes y error Bayes
- Regresión logística
- La regla de vecinos más próximos

Introducción

Disponemos de una muestra de casos que pertenecen a 2 grupos o más. En cada caso se observa una variable vectorial.

Objetivo

Encontrar un buen criterio para asignar un nuevo caso a uno de los grupos (**obtener una regla de clasificación**).

Diferentes nombres para el mismo problema: *supervised classification, statistical learning, discrimination, machine learning, pattern recognition*, etc.

Referencias más importantes

- Devroye, Györfi y Lugosi (1996). *A Probabilistic Theory of Pattern Recognition*. Springer.
- Hastie, Tibshirani y Friedman (2001). *The Elements of Statistical Learning*. Springer.

Planteamiento general

Sean $D_n := (X_1, Y_1), \ldots, (X_n, Y_n)$ vaiid con valores en $\mathbb{R}^p \times \{0, 1\}$.

Objetivo

Dada una nueva observación (X, Y), independiente de las anteriores pero con la misma distribución y de la que solo se conoce X, predecir el valor de Y.

Elementos del problema

- Probabilidades a priori: $\pi_1 = \mathrm{P}(Y = 1)$ y $\pi_0 = \mathrm{P}(Y = 0) = 1 - \pi_1$
- Probabilidad a posteriori de $Y = 1$:

$$\eta(x) = \mathrm{P}(Y = 1 | X = x) = \mathrm{E}(Y | X = x).$$

- Distribución marginal de X: $\mathrm{P}_X(A) = \mathrm{P}(X \in A)$.
- Distribuciones condicionadas: $\mathrm{P}_0(A) = \mathrm{P}(X \in A | Y = 0)$ y $\mathrm{P}_1(A) = \mathrm{P}(X \in A | Y = 1)$.

Error de clasificación

- Una regla de clasificación es una función (medible) $g:\ \mathbb{R}^d \to \{0,1\}$ tal que a cada $x \in \mathbb{R}^d$ le asigna una de las dos clases $g(x) = g(x; D_n)$
- Probabilidad de error de clasificación

$$L_n = \mathrm{P}(g(X) \neq Y | D_n)$$

- Observación: L_n es una variable aleatoria
- También se puede promediar para las distintas muestras de entrenamiento

$$\mathrm{E}(L_n) = \mathrm{P}(g(X) \neq Y)$$

Estimadores de la probabilidad de error

- Error empírico o tasa de error aparente

$$\hat{L}_n = \frac{1}{n} \sum_{i=1}^{n} I_{\{g(x_i) \neq y_i\}}$$

- Tasa de error por validación cruzada (*leave-one-out*)

$$\hat{L}_n^{(vc)} = \frac{1}{n} \sum_{i=1}^{n} I_{\{g_{(-i)}(x_i) \neq y_i\}}$$

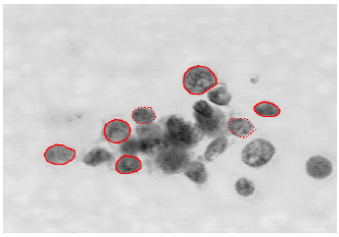
- Uso de una muestra de test $(X_{n+1}, Y_{n+1}), \ldots, (X_{n+m}, Y_{n+m})$ independiente de la muestra de entrenamiento

$$\hat{L}_{n,m} = \frac{1}{m} \sum_{j=1}^{m} I_{\{g(x_{n+j}) \neq y_{n+j}\}}$$

Ejemplo: diagnóstico por imagen

Mediante una punción con aguja fina se extrae una muestra de tejido sospechoso. La muestra se tiñe para resaltar los núcleos de las células.

Las variables corresponden a los valores medios de distintos aspectos de la forma de los núcleos.



El fichero Wisc.RData contiene el `data.frame` `Wisconsin` cuyas variables son:

- Las 10 variables explicativas medidas en pacientes cuyos tumores fueron diagnosticados posteriormente.
- La variable `tipo` que contiene el tipo de tumor (benigno o maligno).

Más información sobre los datos en esta web.

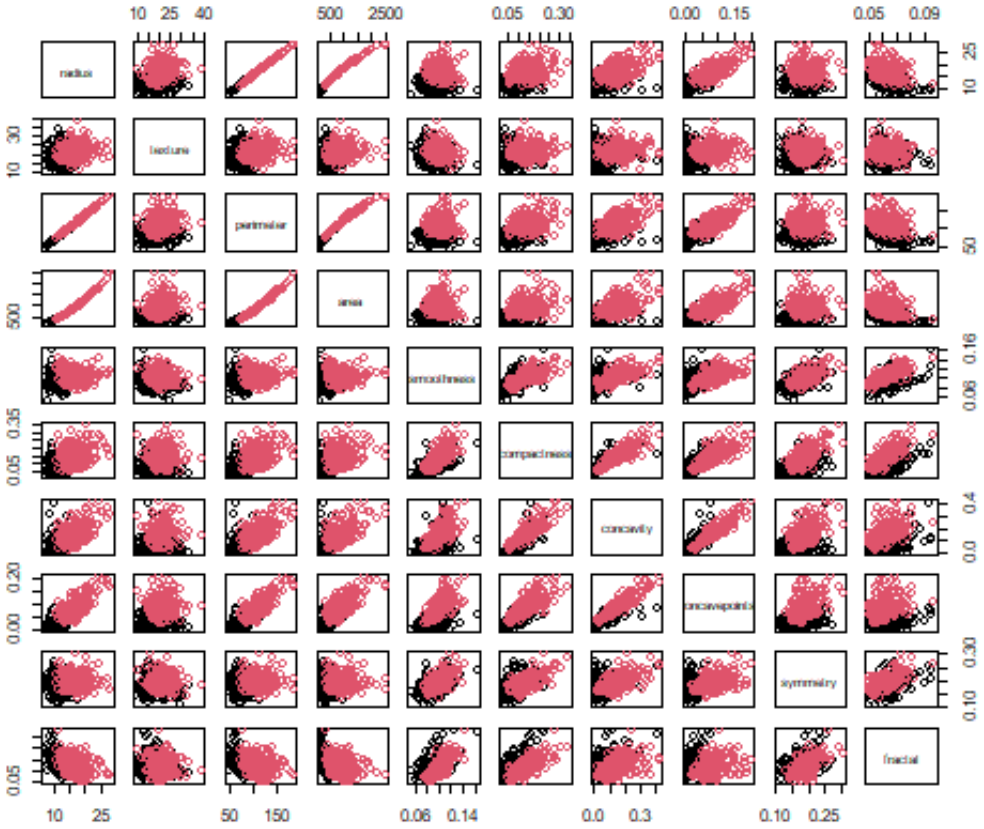
Ejemplo: diagnóstico por imagen

```
load(url('http://verso.mat.uam.es/~joser.berrendero/datos
knitr::kable(head(Wisconsin[1:6]), format = 'html')
```

radius	texture	perimeter	area	smoothness	compactnes
13.540	14.36	87.46	566.3	0.09779	0.0812
13.080	15.71	85.63	520.0	0.10750	0.1270
9.504	12.44	60.34	273.9	0.10240	0.0649
13.030	18.42	82.61	523.8	0.08983	0.0376
8.196	16.84	51.71	201.9	0.08600	0.0594
12.050	14.63	78.04	449.3	0.10310	0.0909

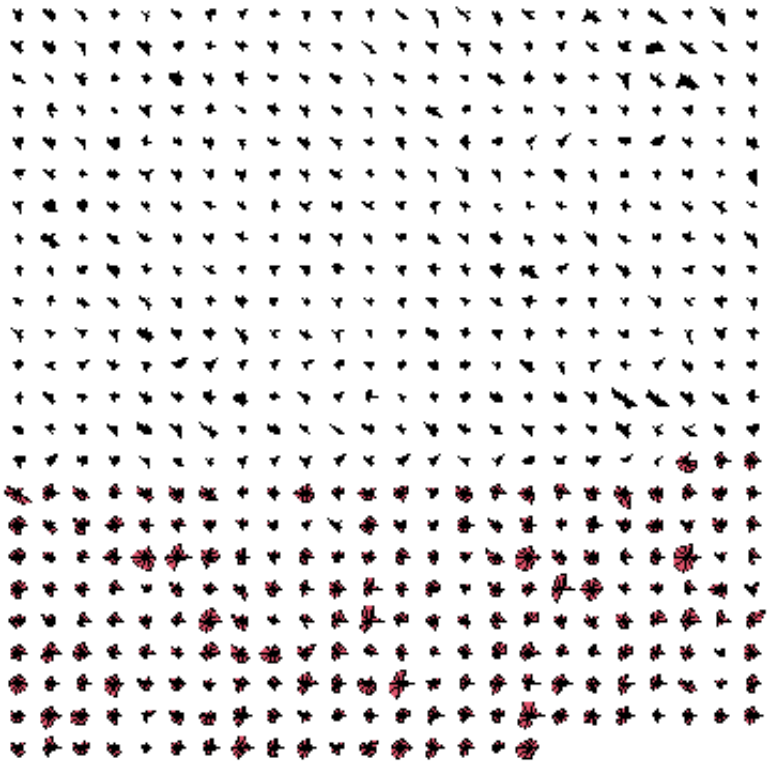
Diagramas de dispersión

```
pairs(Wisconsin[,-11], col=Wisconsin$tipo)
```



Diagramas de estrella

```
stars(Wisconsin[-11], col.stars=Wisconsin$tipo, labels=NU
```

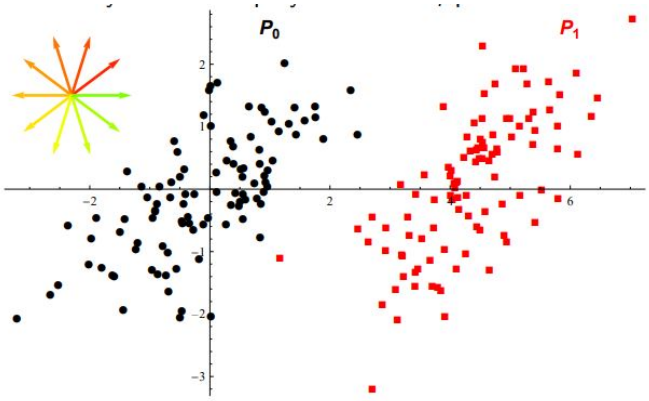


Regla lineal de Fisher

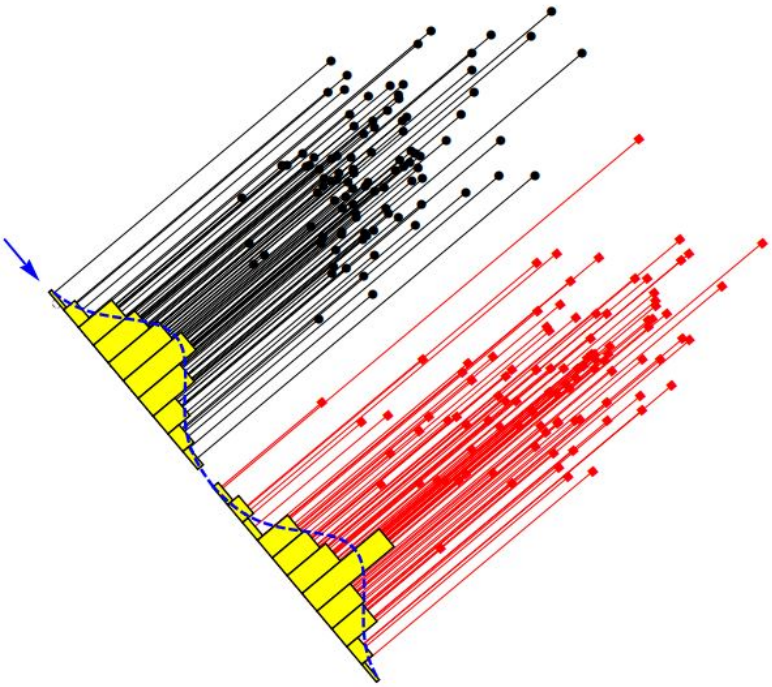
Sean μ_0 y μ_1 vectores de medias de X bajo P_0 y P_1

Homocedasticidad: las matrices de covarianzas de X bajo P_0 y P_1 verifican $\Sigma_0 = \Sigma_1 = \Sigma$.

La idea de Fisher: Proyectar los datos en la dirección más conveniente, a, y utilizar las proyecciones $a'x_i$ para discriminar.

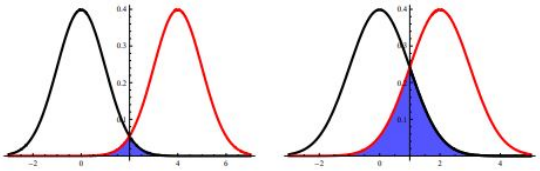


Regla lineal de Fisher

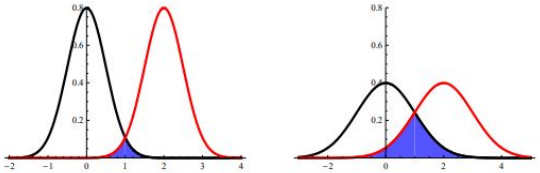


Regla lineal de Fisher

Una buena dirección debe separar lo máximo posible los centros de los grupos. La distancia entre las medias proyectadas $(a'\mu_0 - a'\mu_1)^2 = a'Ba$, donde $B = (\mu_0 - \mu_1)(\mu_0 - \mu_1)'$, debe ser grande.



La varianza de las proyecciones dentro de los grupos $a'\Sigma a$ debe ser lo menor posible.



Regla lineal de Fisher

El problema

Encontrar la dirección a que maximiza

$$f(a) = \frac{a'Ba}{a'\Sigma a} \qquad \text{(cociente de Rayleigh)}$$

La solución

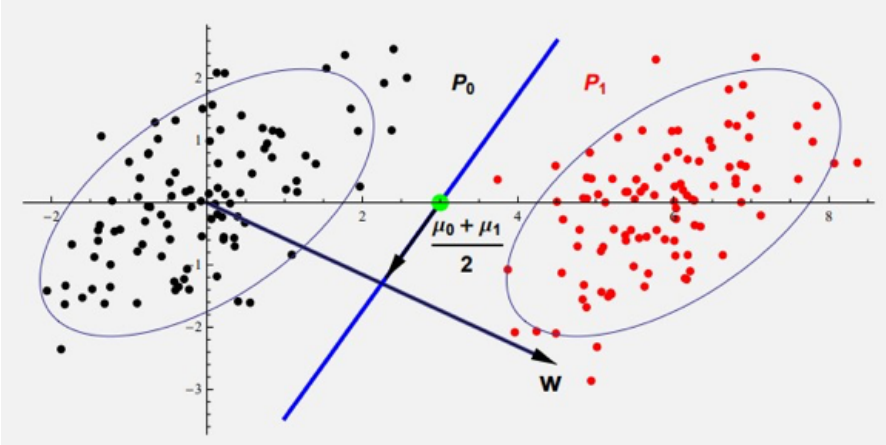
Para cualquier $\lambda \neq 0$, $f(\lambda a) = f(a)$, por lo que es necesario imponer alguna condición de normalización. Es frecuente elegir una solución tal que $a'\Sigma a = 1$ (varianza 1)

- La solución es proporcional al vector discriminante $w = \Sigma^{-1}(\mu_1 - \mu_0)$
- La **función canónica discriminante**, $\tilde{w}$, es la solución del problema de Fisher que verifica $\tilde{w}'\Sigma\tilde{w} = 1$

Regla lineal de Fisher

x se clasifica en P_0 si

$$w'(x-\bar{\mu})<0, \quad \bar{\mu}=(\mu_0+\mu_1)/2 \Leftrightarrow |w'x-w'\mu_0| < |w'x-$$



Estimación de los parámetros

En la práctica no conocemos los parámetros de los modelos, por lo que se sustituyen por sus estimadores naturales.

Para estimar las esperanzas se utilizan las medias muestrales:

$$\hat{\mu}_0 = \bar{x}_0, \quad \hat{\mu}_1 = \bar{x}_1.$$

Para estimar las matrices de covarianzas se utilizan las matrices de covarianzas muestrales:

$$\hat{\Sigma}_0 = S_0, \quad \hat{\Sigma}_1 = S_1.$$

En el caso homocedástico, para estimar Σ se utiliza el estimador combinado

$$S := \frac{n_0 - 1}{n_0 + n_1 - 2} S_0 + \frac{n_1 - 1}{n_0 + n_1 - 2} S_1.$$

Ejemplo

El comando básico es `lda` del paquete `MASS`. Tiene tres argumentos principales:

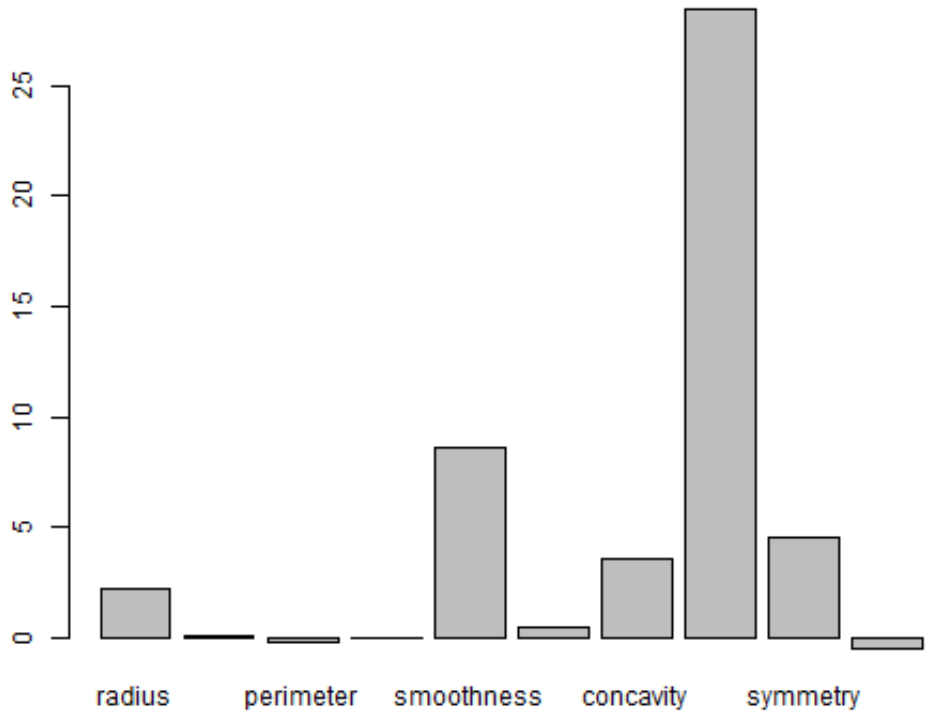
- la expresión que determina la variable que queremos predecir (el tipo de tumor) y qué variables vamos a utilizar,
- el *data frame* que contiene los datos,
- el vector de probabilidades a priori de cada clase

```
resultado.lda <- lda(tipo ~ ., data=Wisconsin, prior = c(
resultado.lda
```

```
## Call:
## lda(tipo ~ ., data = Wisconsin, prior = c(0.5, 0.5))
##
## Prior probabilities of groups:
## benigno maligno
##     0.5     0.5
##
## Group means:
##           radius  texture perimeter     area smoothness compactn
## benigno 12.14652 17.91476  78.07541 462.7902 0.09247765  0.08008
## maligno 17.46283 21.60491 115.36538 978.3764 0.10289849  0.14518
##         concavepoints symmetry    fractal
## benigno    0.02571741 0.174186 0.06286739
## maligno    0.08799000 0.192909 0.06268009
##
## Coefficients of linear discriminants:
##                       LD1
## radius         2.173832578
## texture        0.097479319
## perimeter     -0.243883158
## area          -0.004235635
## smoothness     8.610211091
## compactness    0.431476344
## concavity      3.592356858
## concavepoints 28.529778564
## symmetry       4.489073661
## fractal       -0.529214778
```

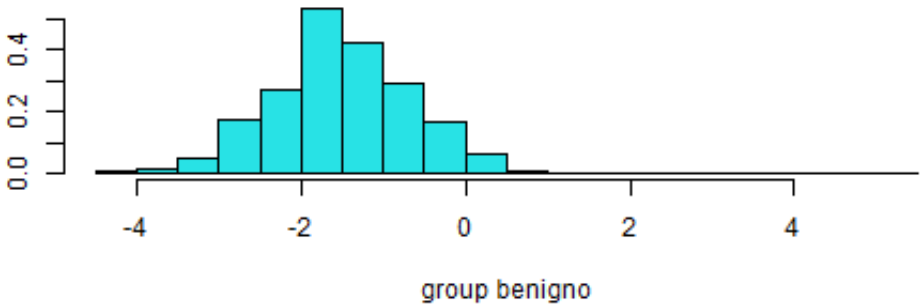
Ejemplo

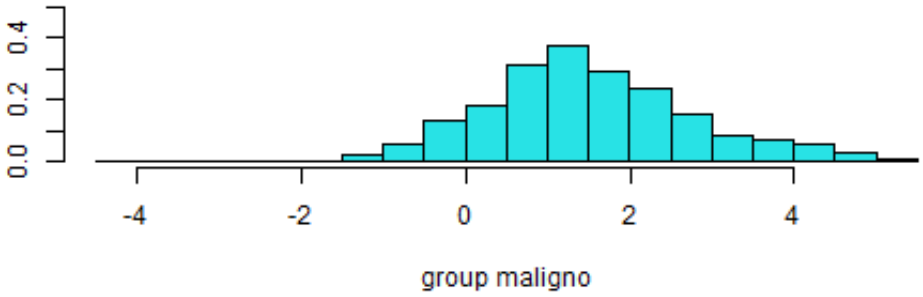
```
barplot(as.vector(resultado.lda$scaling), names.arg = nam
```



Ejemplo

```
plot(resultado.lda)
```





Ejemplo

Predicciones y tasa de error

```
predicciones.lda <- predict(resultado.lda)$class
table(Wisconsin$tipo, predicciones.lda)
```

```
##           predicciones.lda
##            benigno maligno
##   benigno      345       12
##   maligno       22      190
```

```
mean(Wisconsin$tipo != predicciones.lda)
```

```
## [1] 0.05975395
```

Tasa de error por validación cruzada

```
predicciones.lda.cv <- lda(tipo ~ ., data = Wisconsin, pr
mean(Wisconsin$tipo != predicciones.lda.cv)
```

```
## [1] 0.06326889
```

Ejemplo

Para clasificar nuevos vectores de observaciones se usa `predict`:

- Las nuevas observaciones deben estar en un *data frame*
- Los nombres de las variables deben ser los mismos que en la muestra de entrenamiento

```
# Predicciones de dos vectores generados aleatoriamente
x1 <- rnorm(10)
x2 <- rnorm(10)
nuevas.obs <- data.frame(rbind(x1, x2))  # en un data fra
names(nuevas.obs) <- names(Wisconsin[1:10]) # con los mis

predict(resultado.lda, nuevas.obs)$class    # resultado c
```

```
## [1] benigno benigno
## Levels: benigno maligno
```

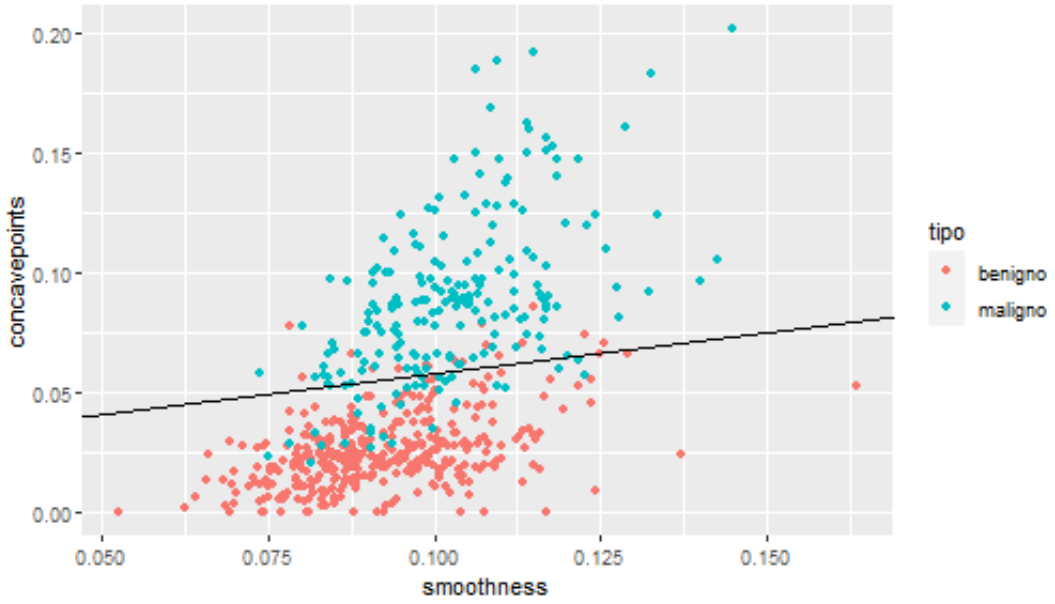
```
predict(resultado.lda, nuevas.obs)$x        # puntuacione
```

```
##           LD1
## x1 -84.89791
## x2 -16.17256
```

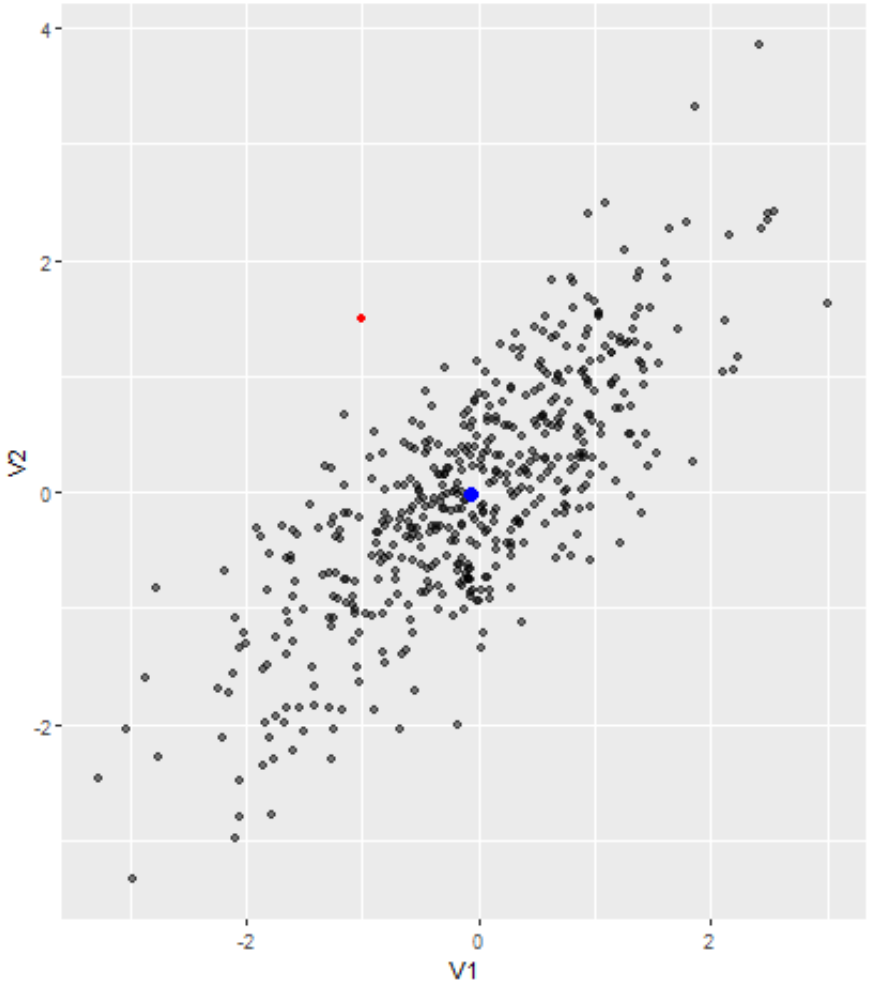
Ejercicio

En la función discriminante de Fisher, las variables `smoothness` y `concavepoints` son las que reciben una mayor ponderación. Repite los cálculos teniendo en cuenta únicamente estas dos variables.

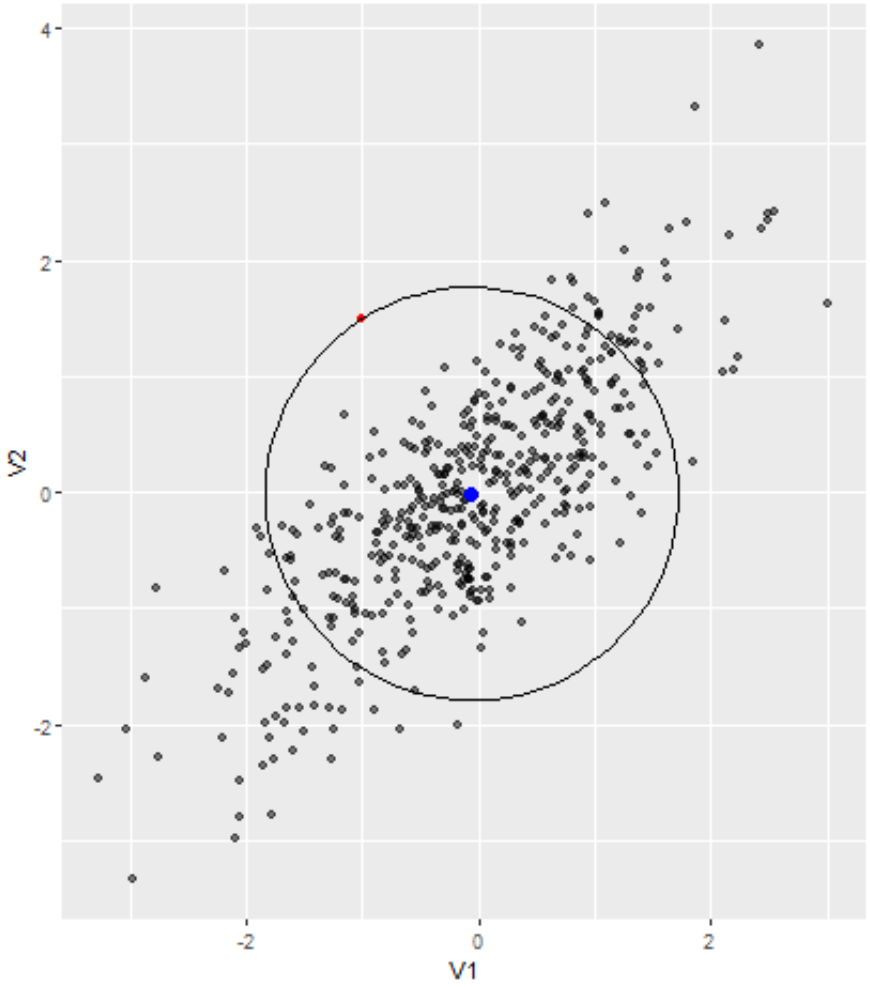
- ¿Cuál es la función discriminante lineal de Fisher en este caso?
- ¿Cuál es la nueva tasa de error usando validación cruzada?



Distancia de Mahalanobis



Distancia de Mahalanobis



Distancia de Mahalanobis

Si X es un vector aleatorio con media μ y matriz de covarianzas Σ, la distancia de Mahalanobis entre X y μ es

$$d_M(X,\mu) = \sqrt{(X-\mu)'\Sigma^{-1}(X-\mu)}.$$

Diagonalizamos $\Sigma = V\Lambda V'$ y definimos $\Sigma^{1/2} = V\Lambda^{1/2}V'$

Entonces:

$$d_M(X,\mu) = ||\Sigma^{-1/2}(X-\mu)||_2 = ||V\Lambda^{-1/2}V'(X-\mu)||_2$$

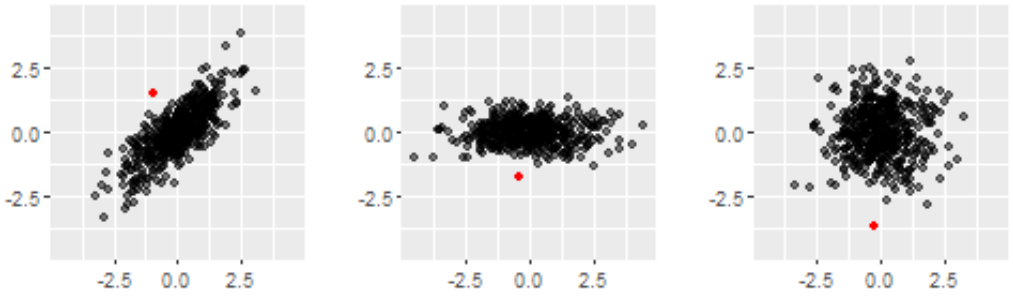
Más explícitamente:

$$d_M(X,\mu) = \left[\frac{((X-\mu)'v_1)^2}{\lambda_1} + \cdots + \frac{((X-\mu)'v_p)^2}{\lambda_p}\right]^{1/2}$$

La distancia de Mahalanobis entre una observación y la media es la norma euclídea del vector de sus componentes principales estandarizadas.

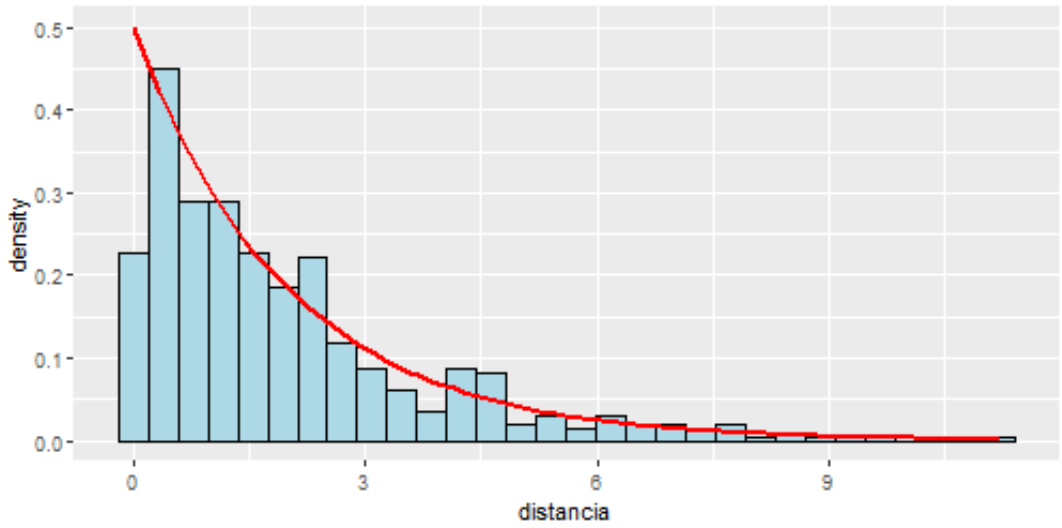
Distancia de Mahalanobis

- Trasladar los datos de forma que su nueva media sea el origen de coordenadas
- Rotarlos para que las correlaciones entre las variables sean todas iguales a cero
- Estandarizar el resultado del paso anterior para que todas las varianzas valgan uno
- Calcular la distancia (euclídea) al origen de los puntos resultantes



Ejemplo

```
n <- 500
datos <- as.data.frame(mvrnorm(n, mu = c(0,0), Sigma = ma
media <- c(mean(datos$V1), mean(datos$V2))
covarianza <- cov(datos)
datos_dist <- datos %>%
  mutate(distancia = mahalanobis(datos, media, covarianza

ggplot(datos_dist) +
  geom_histogram(aes(x = distancia, y = ..density..), fil
  geom_function(fun = 'dchisq', args = list(df = 2), col
```



Regla de Mahalanobis

X se clasifica en P_0 si

$$(X-\mu_0)'\Sigma_0^{-1}(X-\mu_0) < (X-\mu_1)'\Sigma_1^{-1}(X-\mu_1)$$

La frontera que separa P_1 y P_0 es la función cuadrática $Q(x)=0$, donde

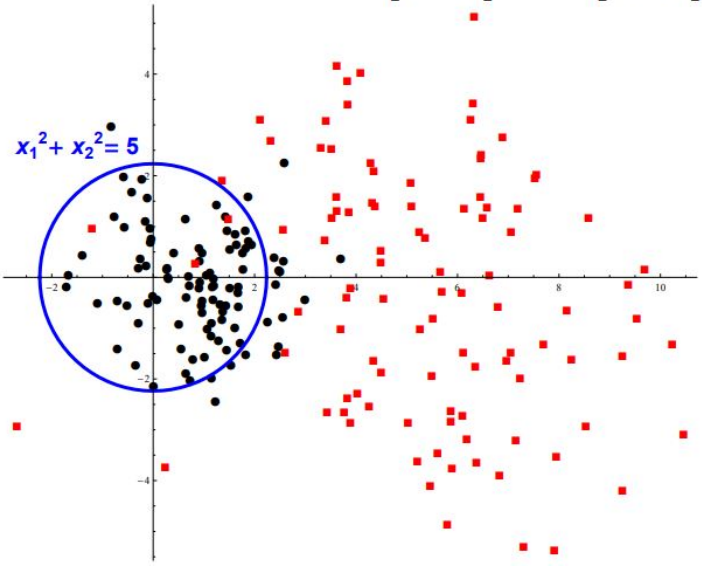
$$Q(x) = x'Ax + 2a'x + c$$

- $A = \Sigma_0^{-1} - \Sigma_1^{-1}$
- $a = \Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0$
- $c = \mu_0'\Sigma_0^{-1}\mu_0 - \mu_1'\Sigma_1^{-1}\mu_1$

¿Qué ocurre si $\Sigma_0 = \Sigma_1$?

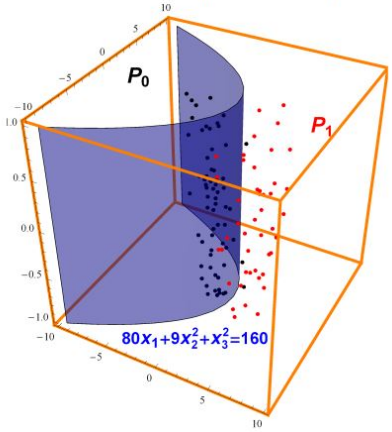
Regla de Mahalanobis

$$\mu_0 = (1,0)', \mu_1 = (5,0)', \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}.$$



Regla de Mahalanobis

$$\mu_0 = (0,0,0)', \ \mu_1 = (4,0,0)', \ \Sigma_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \ \Sigma_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$



Regla y error Bayes

Como función de η

Si $\eta(x) = \mathrm{P}(Y = 1 | X = x)$ la regla Bayes se define como

$$g^*(x) = \begin{cases} 1, & \text{si } \eta(x) > 1/2, \\ 0, & \text{si } \eta(x) \leq 1/2 \end{cases}$$

Como función de f_0 y f_1

Si las distribuciones condicionadas P_0 y P_1 son continuas con densidades f_0 y f_1, y $\pi_0 = \mathrm{P}(Y = 0)$, $\pi_1 = \mathrm{P}(Y = 1)$ son las probabilidades a priori,

$$\eta(x) = \frac{f_1(x)\pi_1}{f_1(x)\pi_1 + f_0(x)\pi_0}$$

Por lo tanto,

$$g^*(x) = \begin{cases} 1, & \text{si } \pi_1 f_1(x) > \pi_0 f_0(x), \\ 0, & \text{si } \pi_1 f_1(x) \leq \pi_0 f_0(x) \end{cases}$$

Error Bayes

El error de g^* se llama error Bayes

$$L^* = \mathrm{P}(g^*(X) \neq Y)$$

Proposición (la regla Bayes es la regla óptima).

Para cualquier regla de clasificación g, se cumple $L^* \leq L(g)$.

Demostración: Dado $X = x$,

$$\mathrm{P}(g(X) \neq Y | X = x) = \eta(x) - \mathbb{I}_{\{g(x)=1\}}[2\eta(x) - 1]$$

$$\begin{aligned} &\mathrm{P}(g(X) \neq Y | X = x) - \mathrm{P}(g^*(X) \neq Y | X = x) \\ &= [2\eta(x) - 1][\mathbb{I}_{\{g^*(x)=1\}} - \mathbb{I}_{\{g(x)=1\}}] \geq 0 \end{aligned}$$

Se toman esperanzas respecto a X.

Formas alternativas del error Bayes

Como función de η

$$L^* = \mathrm{E}(\min\{\eta(X), 1-\eta(X)\}) = \frac{1}{2} - \frac{1}{2}\mathrm{E}(|2\eta(X)-1|)$$

- $L^* \leq 1/2$
- Si para muchos valores de X una de las clases es mucho más probable que la otra, L^* es pequeño.

Como función de f_0 y f_1 (suponiendo $\pi_0 = \pi_1 = 1/2$)

$$L^* = \frac{1}{2}\int \min\{f_1(x), f_0(x)\}dx = \frac{1}{2} - \frac{1}{4}\int |f_0(x) - f_1(x)$$

Consistencia de una regla de clasificación

Un clasificador g_n es **consistente** para cierta distribución de (X, Y) si $\mathrm{E}(L_n) \to L^*$ cuando $n \to \infty$

Un clasificador g_n es **fuertemente consistente** para cierta distribución de (X, Y) si $L_n \to L^*$ c.s. cuando $n \to \infty$

Un clasificador g_n es **universalmente (fuertemente) consistente** cuando es (fuertemente) consistente para cualquier distribución de (X, Y).

g_n consistente $\Leftrightarrow L_n \to L^*$, en probabilidad, cuando $n \to \infty$.

Métodos para construir clasificadores

- **Reglas plug-in:** estimar $\eta(x)$ mediante $\hat{\eta}(x)$ y sustituir $\eta(x)$ por $\hat{\eta}(x)$ en la definición de la regla de Bayes g^*.
  - Puede suponerse que $\eta(x)$ es conocida salvo un número finito de parámetros y estimarlos.
  - Se puede estimar $\eta(x)$ de forma no paramétrica.
  - En lugar de estimar $\eta(x)$, a veces se estima π_0, π_1, f_0 y f_1. La estimación de f_0 y f_1 también puede ser paramétrica o no paramétrica.
- **Minimización del riesgo empírico:** se obtienen reglas que minimizan $\widehat{L}_n$ en una familia apropiada $\mathcal{G}$.

Regla Bayes bajo normalidad

$$f(x) = c\,|\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right\}, \quad x \in \mathbb{R}^p.$$

Caso heterocedástico

$$g^*(x) = 1 \Leftrightarrow d_M^2(x,\mu_0) > d_M^2(x,\mu_1) + 2\log\frac{\pi_0|\Sigma_1|}{\pi_1|\Sigma_0|}$$

Salvo una constante coincide con la regla de Mahalanobis

Caso homocedástico

$\Sigma_0 = \Sigma_1 = \Sigma.$

$$g^*(x) = 1 \Leftrightarrow w'(x-\bar{\mu}) > \log(\pi_0/\pi_1)$$

Coincide con la regla de Fisher trasladada en función de las probabilidades a priori

Ejemplo

```
resultado.qda <- qda(Wisconsin$tipo ~ ., data = Wisconsin
predicciones.qda <- predict(resultado.qda)$class
table(Wisconsin$tipo, predicciones.qda)
```

```
##           predicciones.qda
##            benigno maligno
##   benigno      345      12
##   maligno       21     191
```

```
mean(predicciones.qda != Wisconsin$tipo)
```

```
## [1] 0.05799649
```

```
# Validación cruzada
resultado.qda.cv <- qda(Wisconsin$tipo ~ .,
                    data = Wisconsin,
                    prior=c(0.5,0.5),
                    CV=TRUE)
predicciones.qda.cv <- resultado.qda.cv$class
mean(predicciones.qda.cv != Wisconsin$tipo)
```

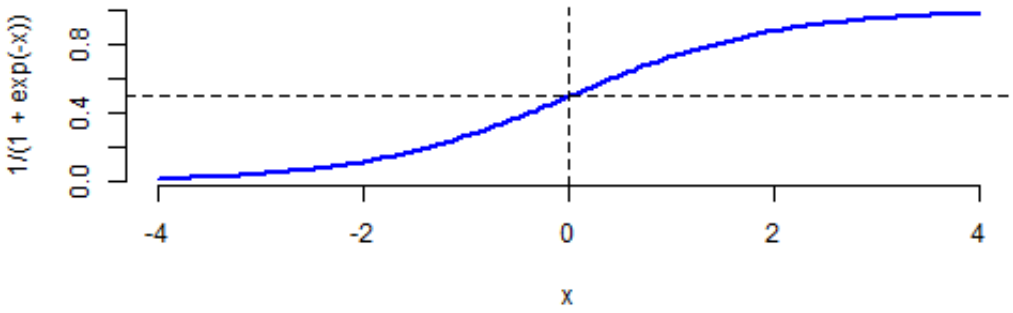
```
## [1] 0.06678383
```

Regresión logística

Las variables $Y_1, \ldots, Y_n$ son independientes y tienen distribución de Bernoulli. La probabilidad de *éxito* depende de las variables regresoras.

$$\eta(x_i) := \eta_i = f(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{id}) = \frac{}{1 + e^{-\beta_0}}$$

donde $f(x) = 1/(1 + e^{-x})$ es la *función logística*.



Algunas propiedades de la función logística

- $f(0) = 1/2$
- $f(-x) = 1 - f(x)$
- $f'(x) = f(x)(1 - f(x))$

La función logística no es la única que se ha utilizado para modelizar este tipo de datos.

El modelo **probit** consiste en suponer $\eta_i = \Phi(\beta' x_i)$, donde Φ es la función de distribución normal estándar.

Poblaciones normales y modelo logístico

Notación. Para $i = 1, \ldots, n$, cada observación está formada por un vector de variables regresoras $x_i = (1, x_{i1}, \ldots, x_{ip})'$ y el valor de la variable respuesta y_i.

Hipótesis. Suponemos que P_0 es $\mathrm{N}(\mu_0, \Sigma)$ y P_1 es $\mathrm{N}(\mu_1, \Sigma)$. Además $\pi_0 = \pi_1$.

Bajo las condiciones anteriores se verifica el modelo logístico, ya que

$$\log \frac{\eta(x_i)}{1 - \eta(x_i)} = \beta_0 + \beta_1 x_{i1} + \cdots \beta_d x_{id},$$

donde $w := (\beta_1, \ldots, \beta_p)' = \Sigma^{-1}(\mu_1 - \mu_0)$ y $\beta_0 = -w'(\mu_0 + \mu_1)/2$.

La regla de Fisher y la regla logística coinciden en términos poblacionales pero difieren en el método de estimación de los parámetros

Interpretación de los parámetros

Llamamos O_i a la **razón de probabilidades** para la observación i:

$$O_i = \frac{\eta_i}{1-\eta_i}$$

¿Cómo se interpreta el valor de O_i? ¿Qué significa, por ejemplo, $O_i = 2$?

Si se cumple el modelo de regresión logística, entonces

$$O_i = e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{id}}$$

¿Cómo varía la razón de probabilidades si la variable regresora x_{ij} se incrementa una unidad?

$$\frac{O_i'}{O_i} = \frac{e^{\beta_0+\cdots+\beta_j(x+1)+\cdots+\beta_p x_{id}}}{e^{\beta_0+\cdots+\beta_j x+\cdots+\beta_p x_{id}}} = e^{\beta_j}.$$

Por tanto e^{β_j} es la variación de la razón de probabilidades cuando la variable regresora j se incrementa en una unidad y el resto de variables permanece constante.

Estimación

Para estimar los parámetros se usa el método de máxima verosimilitud.

Por ejemplo, si observamos los siguientes datos (x_i, y_i): $(2, 0), (1, 1), (3, 1)$, entonces $\hat{\beta}_0$ y $\hat{\beta}_1$ son los valores que maximizan la función de verosimilitud

$$L(\beta_0, \beta_1) = \mathrm{P}(Y = 0 \mid x = 2)\mathrm{P}(Y = 1 \mid x = 1)\mathrm{P}(Y = 1 \mid x$$

$$L(\beta_0, \beta_1) = \left(1 - \frac{1}{1 + e^{-\beta_0 - 2\beta_1}}\right)\left(\frac{1}{1 + e^{-\beta_0 - \beta_1}}\right)\left(\frac{1}{1 + e^{-\beta}}\right.$$

Esta función es cóncava.

Se pueden aplicar algoritmos estándar de optimización para maximizarla.

Estimación

$$L(\beta) = \prod_{i=1}^{n} \eta_i^{Y_i} (1-\eta_i)^{1-Y_i}.$$

$$\ell(\beta) = \log L(\beta) = \sum_{i=1}^{n} [Y_i \log \eta_i + (1-Y_i) \log(1-\eta_i)]$$

El EMV es el valor para el que se anula el gradiente:

$$\nabla \ell(\hat{\beta}) = \sum_{i=1}^{n} \left[Y_i x_i - \frac{1}{1+e^{-x_i' \hat{\beta}}} x_i \right] = 0$$

Estas ecuaciones son análogas a las ecuaciones normales en regresión lineal:

$$\sum_{i=1}^{n} (Y_i - \hat{\eta}_i) x_i = 0 \Leftrightarrow X'Y = X'\hat{\eta}.$$

Desviaciones

Las desviaciones (*deviances*) se definen:

$$D_i^2 = -2[Y_i \log \hat{\eta}_i + (1 - Y_i) \log(1 - \hat{\eta}_i)]$$

- Si $Y_i = 1$, ¿cómo cambia D_i^2 cuando $\hat{\eta}_i$ decrece a 0?
- Si $Y_i = 0$, ¿cómo cambia D_i^2 cuando $\hat{\eta}_i$ crece a 1?

Los valores D_i^2 hacen el papel de los residuos en regresión lineal.

El análogo de la SCR es $\sum_{i=1}^n D_i^2$.

Se cumple $D^2 = \sum_{i=1}^n D_i^2 = -2\ell(\hat{\beta})$.

Desviaciones

Para valorar la bondad del ajuste del modelo a los datos se puede usar D^2.

Al igual que en el caso de regrsión lineal hay que tener en cuenta la complejidad del modelo.

Una posibilidad es usar el **criterio de información de Akaike**:

$$\text{AIC} = -2\ell(\hat{\beta}) + 2(d+1) = D^2 + 2(d+1).$$

Inferencia

Aplicando la teoría asintótica de los EMV se demuestra que, si n es suficientemente grande,

$$\hat{\beta} \cong \mathrm{N}_{p+1}(\beta, (X'\hat{W}X)^{-1}),$$

donde $\hat{W} = \mathrm{diag}(\hat{\eta}_1(1-\hat{\eta}_1), \ldots, \hat{\eta}_n(1-\hat{\eta}_n))$.

Esta aproximación es la base de los contrastes e intervalos para los parámetros del modelo.

Estadístico de Wald: si $\beta_j = 0$,

$$\frac{\hat{\beta}_j}{\mathrm{e.t.}(\hat{\beta})} \cong \mathrm{N}(0,1),$$

donde $\mathrm{e.t.}(\hat{\beta})$ es la raíz del elemento correspondiente de la diagonal de $(X'\hat{W}X)^{-1}$.

Un ejemplo con datos simulados

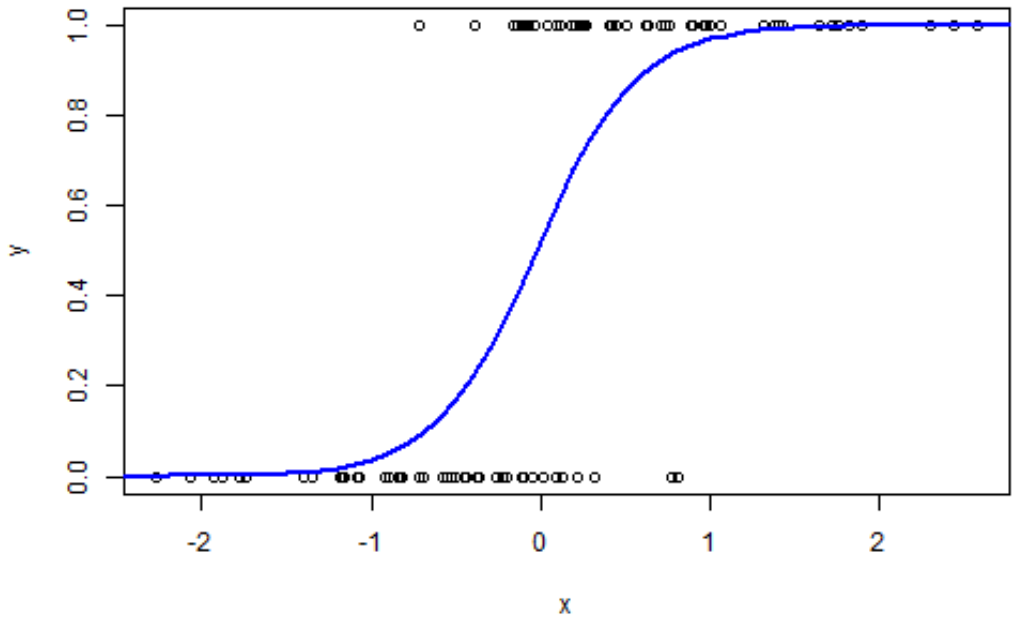
```
set.seed(100)
n <- 100
beta0 <- 0
beta1 <- 3
x <- rnorm(n)  # el modelo no asume normalidad de x
p = 1/(1+exp(-beta0-beta1*x))
y = rbinom(n, 1, p)

# Ajusta el modelo
reg = glm(y~x, family=binomial)
summary(reg)
```

Un ejemplo con datos simulados

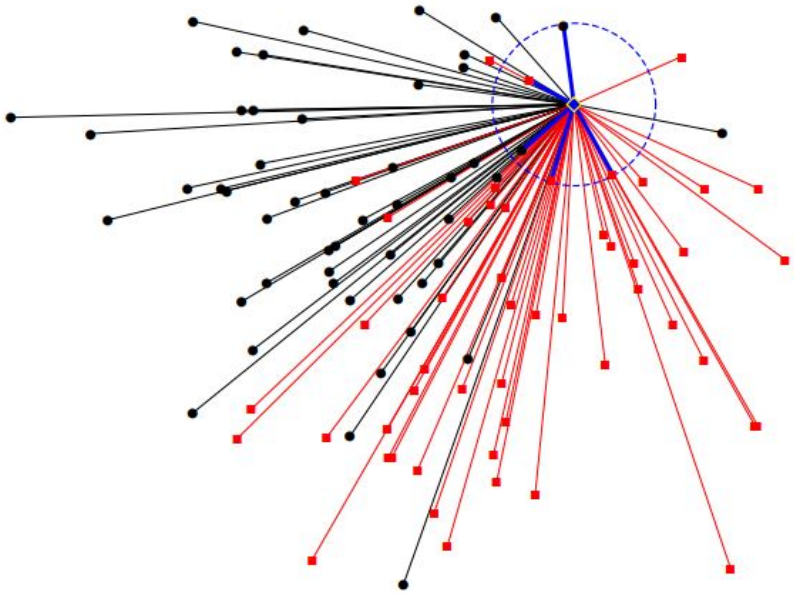
```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.40849  -0.53743  -0.00721   0.48375   2.19983
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.08244    0.29764   0.277    0.782
## x            3.37842    0.72712   4.646 3.38e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 138.629  on 99  degrees of freedom
## Residual deviance:  70.219  on 98  degrees of freedom
## AIC: 74.219
##
## Number of Fisher Scoring iterations: 6
```

Probabilidades estimadas

```
datos <- data.frame(x = seq(-4, 4, 0.1))
probabilidades <- predict(reg, datos, type = 'response')
plot(x, y, pch = 21)
lines(datos$x, probabilidades, col = 'blue', lwd = 2)
```



Regla de vecinos más próximos



Regla de vecinos más próximos

$\eta(x) = \mathrm{P}(Y = 1|X = x)$ se estima mediante la proporción de unos para puntos muestrales x_i que están *cerca* de x.

Sea $k \in \mathbb{N}$ un número prefijado. Se dice que X_i es el k -ésimo vecino más próximo de x si la distancia (euclídea) $\|x - X_i\|$ es la k-ésima más pequeña entre $\|x - X_1\|, \ldots, \|x - X_n\|$.

Regla kNN:

$$g_n(x) = 1 \Leftrightarrow \sum_{i=1}^{n} w_i I_{\{Y_i=1\}} > \sum_{i=1}^{n} w_i I_{\{Y_i=0\}},$$

donde $w_i = 1/k$ si X_i está entre los k puntos muestrales más cercanos a x (es uno de sus k vecinos más próximos) y $w_i = 0$ en caso contrario.

Ejemplo

El comando principal es `knn` del paquete `class`. Tiene cuatro argumentos:

- La matriz o *data frame* con las variables explicativas de entrenamiento (*training*)
- La matriz o *data frame* con las variables que queremos clasificar (*test*)
- El vector que contiene la clase a la que pertenece cada observación del conjunto de entrenamiento
- El número k de vecinos que queremos considerar.

El resultado es un vector con los grupos en los que se clasifican los datos de test.

Ejemplo

Tasa de error aparente

El mismo conjunto de entrenamiento y de test.

```
resultado.knn3 <- knn(Wisconsin[,-11], Wisconsin[,-11], W
error.knn3 <- mean(resultado.knn3 != Wisconsin$tipo)
error.knn3
```

```
## [1] 0.07029877
```

Tasa de error por validación cruzada

Usamos el comando `knn.cv` (igual que `knn` sin datos de test):

```
resultado.knn3.cv <- knn.cv(Wisconsin[,-11], Wisconsin[,1
error.knn3.cv <- mean(resultado.knn3.cv != Wisconsin$tipo
error.knn3.cv
```

```
## [1] 0.1230228
```

Cuestiones

- Calcula la tasa de error aparente utilizando los 4 vecinos más próximos. Repítelo varias veces. ¿Qué se observa?
- Estudia cómo van cambiando las tasas de error aparente y por validación cruzada a medida que aumenta el valor de k.
- Calcula las tasas de error aparente y por validación cruzada si $k = 3$ pero usando únicamente las variables `smoothness` y `concavepoints`.

La regla del vecino más próximo

Supongamos n es grande y usamos la regla del vecino más próximo con $k = 1$. ¿Qué error de clasificación podemos esperar?

Si $(X_{(1)}, Y_{(1)})$ es el vecino más próximo de (X, Y), entonces $X_{(1)} \approx X$ y como consecuencia $Y_{(1)}$ e Y se comportan aproximadamente como vaiid $\mathrm{B}(1, \eta(x))$.

Entonces

$$\mathrm{E}(L_n) \approx L_{1NN} := 2\mathrm{E}(\eta(X)(1 - \eta(X)))$$

Más formalmente, se puede probar que $\lim_{n\to\infty} \mathrm{E}(L_n) = L_{1NN}$, para cualquier distribución de los datos.

Además, se verifica

$$L^* \leq L_{1NN} \leq 2L^*(1 - L^*)$$

Consistencia de la regla knn

Sea L_n la probabilidad de error del clasificador kNN.

Teorema (Stone, 1977) de consistencia universal débil. Si $k_n \to \infty$ y $k_n/n \to 0$, entonces para cualquier distribución (X, Y), se tiene que $\mathrm{E}(L_n) \to L^*$.

Teorema (Devroye y Györfi, 1985) de consistencia universal fuerte. Si $k_n \to \infty$ y $k_n/n \to 0$, entonces para cualquier distribución (X, Y) tal que X tiene densidad se tiene que $L_n \to L^*$ c.s.